



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

PROCESO DE  
COORDINACIÓN DE LAS  
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros  
Informaticos

# ANX-PR/CL/001-01

## GUÍA DE APRENDIZAJE

### ASIGNATURA

105001049 - Experimentación En Ingeniería De Software Con Aprendizaje Profundo

### PLAN DE ESTUDIOS

10CD - Grado En Ciencia De Datos E Inteligencia Artificial

### CURSO ACADÉMICO Y SEMESTRE

2023/24 - Primer semestre

## Índice

---

### Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	2
5. Descripción de la asignatura y temario.....	3
6. Cronograma.....	5
7. Actividades y criterios de evaluación.....	7
8. Recursos didácticos.....	9

## 1. Datos descriptivos

---

### 1.1. Datos de la asignatura

<b>Nombre de la asignatura</b>	105001049 - Experimentación en Ingeniería de Software con Aprendizaje Profundo
<b>No de créditos</b>	3 ECTS
<b>Carácter</b>	Optativa
<b>Curso</b>	Cuarto curso
<b>Semestre</b>	Séptimo semestre
<b>Período de impartición</b>	Septiembre-Enero
<b>Idioma de impartición</b>	Castellano
<b>Titulación</b>	10CD - Grado en Ciencia de Datos e Inteligencia Artificial
<b>Centro responsable de la titulación</b>	10 - Escuela Tecnica Superior De Ingenieros Informaticos
<b>Curso académico</b>	2023-24

## 2. Profesorado

---

### 2.1. Profesorado implicado en la docencia

<b>Nombre</b>	<b>Despacho</b>	<b>Correo electrónico</b>	<b>Horario de tutorías</b> *
Sira Vegas Hernandez (Coordinador/a)		sira.vegas@upm.es	- -

\* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

## 3. Conocimientos previos recomendados

---

### 3.1. Asignaturas previas que se recomienda haber cursado

- Aprendizaje Automático Ii
- Probabilidades Y Estadística I
- Probabilidades Y Estadística Ii
- MÉtodos CIÁsicos Para PredicciÓn

### 3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Lenguaje de programación Python

## 4. Competencias y resultados de aprendizaje

---

### 4.1. Competencias

CB03 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

CB04 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

CB05 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

CE02 - Capacidad de diseñar, implementar y evaluar soluciones algorítmicas eficientes para problemas computacionales de ciencia de datos e inteligencia artificial de acuerdo con los requisitos establecidos.

CE18 - Capacidad de diseñar y construir soluciones basadas en redes de neuronas artificiales para problemas en el ámbito del título, como son los de clasificación y estimación.

CG01 - Capacidad de trabajo en equipo, en entornos interdisciplinarios y complejos, negociando y resolviendo conflictos, diseñando soluciones eficientes, fiables, robustas y responsables.

## 4.2. Resultados del aprendizaje

RA165 - Aprender a diseñar experimentos destinados a evaluar soluciones basadas en redes de neuronas artificiales para problemas relacionados con la Ingeniería de Software

RA167 - Aprender a reportar experimentos destinados a evaluar soluciones basadas en redes de neuronas artificiales para problemas relacionados con la Ingeniería de Software

RA41 - Manejar técnicas básicas de inferencia estadística

RA166 - Aprender a analizar experimentos destinados a evaluar soluciones basadas en redes de neuronas artificiales para problemas relacionados con la Ingeniería de Software

RA163 - Capacidad para leer, comprender e implementar artículos científicos

RA19 - RA-FMA-2 Resolver problemas con ayuda de software matemático.

## 5. Descripción de la asignatura y temario

---

### 5.1. Descripción de la asignatura

La Ingeniería de Software (IS) es el proceso de diseño, desarrollo y mantenimiento de software utilizando principios de ingeniería y métodos sistemáticos. La IS trata de aplicar los principios y técnicas de la ingeniería para crear software de alta calidad y confiable que cumpla con los requisitos y necesidades de los usuarios.

El aprendizaje profundo está emergiendo como un enfoque muy potente para abordar los problemas a los que se enfrentan la IS. Actualmente se utiliza aprendizaje profundo para implementar herramientas de IS que ayudan a:

- Generar código automáticamente.
- Documentar código automáticamente.
- Automatizar la generación de casos de prueba.
- Detectar errores en el código y corregirlos de forma automática.

Los desarrolladores de software tienen a día de hoy a su disposición numerosas herramientas, y necesitan saber cuáles de ellas funcionarán mejor en su entorno. Para ello, la IS se basa en la experimentación, ya que permite

evaluar la efectividad de los métodos, técnicas y herramientas de desarrollo de software, determinando si un enfoque en particular es efectivo o no mediante la recopilación de evidencia empírica.

El objetivo principal de la asignatura es que los alumnos aprendan a evaluar las herramientas basadas en aprendizaje profundo disponibles en IS, siguiendo la metodología de diseño y análisis de experimentos.

## 5.2. Temario de la asignatura

### 1. Introducción

#### 1.1. Ingeniería de Software

#### 1.2. El papel del aprendizaje profundo en la ingeniería de software

#### 1.3. Experimentación en Ingeniería de Software con Aprendizaje Profundo

### 2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo

#### 2.1. Formulación de hipótesis

#### 2.2. Selección de variables

#### 2.3. Elección de diseño

### 3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo

#### 3.1. Instrumentación

#### 3.2. Análisis de los resultados experimentales

#### 3.3. Interpretación de resultados

#### 3.4. Evaluación de la validez

## 6. Cronograma

### 6.1. Cronograma de la asignatura \*

Sem	Actividad en aula	Actividad en laboratorio	Tele-enseñanza	Actividades de evaluación
1	<b>1. Introducción</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
2	<b>1. Introducción</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
3	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
4	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
5	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			<b>Práctica: Entrega 1</b> TG: Técnica del tipo Trabajo en Grupo Evaluación continua No presencial Duración: 10:00
6	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
7	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas			
8	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas			
9	<b>2. Diseñando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas			<b>Práctica: Entrega 2</b> TG: Técnica del tipo Trabajo en Grupo Evaluación continua No presencial Duración: 10:00
10	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			

11	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
12	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
13	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
14	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas			
15	<b>3. Analizando Experimentos en Ingeniería de Software para Aprendizaje Profundo</b> Duración: 02:00 AC: Actividad del tipo Acciones Cooperativas			<b>Práctica: Entrega 3</b> TG: Técnica del tipo Trabajo en Grupo Evaluación continua No presencial Duración: 10:00
16				<b>Presentación de la práctica</b> PG: Técnica del tipo Presentación en Grupo Evaluación continua Presencial Duración: 02:00
17				<b>Re-presentación de la práctica</b> PG: Técnica del tipo Presentación en Grupo Evaluación sólo prueba final Presencial Duración: 02:00  <b>Práctica: Todas las re-entregas</b> TG: Técnica del tipo Trabajo en Grupo Evaluación sólo prueba final No presencial Duración: 30:00

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

\* El cronograma sigue una planificación teórica de la asignatura y puede sufrir modificaciones durante el curso derivadas de la situación creada por la COVID-19.

## 7. Actividades y criterios de evaluación

### 7.1. Actividades de evaluación de la asignatura

#### 7.1.1. Evaluación (progresiva)

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
5	Práctica: Entrega 1	TG: Técnica del tipo Trabajo en Grupo	No Presencial	10:00	10%	3 / 10	CB03 CG01
9	Práctica: Entrega 2	TG: Técnica del tipo Trabajo en Grupo	No Presencial	10:00	40%	3 / 10	CG01 CE02 CE18
15	Práctica: Entrega 3	TG: Técnica del tipo Trabajo en Grupo	No Presencial	10:00	40%	3 / 10	CG01 CE02 CE18
16	Presentación de la práctica	PG: Técnica del tipo Presentación en Grupo	Presencial	02:00	10%	3 / 10	CB04 CB05 CG01

#### 7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Re-presentación de la práctica	PG: Técnica del tipo Presentación en Grupo	Presencial	02:00	10%	5 / 10	CB04 CB05 CG01
17	Práctica: Todas las re-entregas	TG: Técnica del tipo Trabajo en Grupo	No Presencial	30:00	90%	5 / 10	CB03 CG01 CE02 CE18

#### 7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Re-presentación de la práctica	PG: Técnica del tipo Presentación en Grupo	Presencial	02:00	10%	5 / 10	CB04 CB05 CG01
Práctica: Todas las re-entregas	TG: Técnica del tipo Trabajo en Grupo	Presencial	30:00	90%	5 / 10	

## 7.2. Criterios de evaluación

### Evaluación progresiva:

Los alumnos serán evaluados en función de la nota obtenida en las distintas entregas de la práctica. No habrá examen. La nota final se calculará utilizando la media ponderada (como se indica más arriba).

### Evaluación global:

Cuando la nota obtenida en la evaluación progresiva sea menor de 5, los alumnos deberán re-entregar las entregas de la práctica suspensas (y/o realizar de nuevo la presentación). El cálculo de la nota final se hará del mismo modo a la evaluación progresiva.

### Convocatoria extraordinaria:

Cuando la nota obtenida en la evaluación global sea menor de 4, los alumnos deberán re-entregar las entregas de la práctica suspensas (y/o realizar de nuevo la presentación). El cálculo de la nota final se hará del mismo modo a la evaluación progresiva.

## 8. Recursos didácticos

---

### 8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Claes Wohlin et al. Experimentation in software engineering: an introduction. Kluwer 2000	Bibliografía	
Natalia Juristo, Ana Moreno. Basics of Software Engineering Experimentation. Kluwer 2001	Bibliografía	